REMARKS

I would like to thank the Examiner for taking the time to talk to me on the telephone on Thursday, November 16. I understand from his comments that I should have got everything right the first time, and am sorry for testing his patience. I can assure the Examiner, and the Commissioner for Patents, that I am genuinely trying to advance the prosecution of this case.

I recognize that my patent application is quite long (because I wanted to ensure an enabling disclosure). I wonder whether some of the Examiner's reservations about the invention are related to not seeing where the invention finds use in the real world: what is the point of the invention? What good does it do? Why would I want to buy a database system which was constructed as claimed/which operated as claimed?

I have in the patent application specifically discussed some real world examples of the areas where the invention is useful. The patent paragraph numbers, according to the specification I see when I view the USPTO PAIR version of the application that the Examiner should review are, as a non-exhaustive list:-
37, 39, 41, 42, 77, 147, 150, 151, 310, 433, and especially the "Some Benefits of the invention" section from paragraph 735 to 818 – an entire section discussing examples of real world advantages of the claimed invention.

The claims require insert-only functionality with no alteration of timestamps for start in valid time and end in valid time – earlier summary positions are still present in memory, so that queries can really return the answer they would give if the database was really queried at a selected point in transaction/system time. This is specified in each independent claim and is not in the prior art.

I have also produced a document entitled "The Point Is" which now follows. I will then address the Examiner's actual objections. The purpose of the "The Point Is" part of this response is to assist the Examiner in seeing what it is that my invention give the world that it did not have, and why it matters.

The Point Is

This document describes a logical progression of the elements (comprising the patent application), which explain the point of the invention and identifies the benefits that the invention offers. I set out a contextual framework, within which one can place the ideas and concepts contained within the patent application. For clarity I refer to paragraphs within in the patent text where appropriate.

The theory of Temporal Databases has existed for at least 20 years. Concepts such as Transaction Time, Valid Time, Chronons and the 'moving point now' are generally recognised terms. However even though many papers have described the potential advantages of such a technology, the theory has remained a largely academic subject, not least because the theory has lacked a practical, implementable model or design upon which to build these concepts.

To date there have been attempts at temporal databases but these have used different approaches to that contained within this patent application and crucially are unable to address the complexities of delivering a robust solution. The fact that no widely available database technology yet exists for temporal data management is proof that the second hurdle has not yet been successfully resolved. This patent application is an inventive solution which solves the problem of how to handle time in bi-temporal databases. Temporal Normal Form (TMNF), also called the Summary Position, and the way it is stored in the database, is the practical model which is different to prior art and critically is a model the industry needs for temporal data management. The patent submitted is the specification for TMNF and is written in such a way that a competent developer could implement code to create a temporal database.

(Paragraphs discussing this in my patent application are paragraghs [0016] [0017] [0018] [0020] [0021] [0022] [0023] [0024] [0025] [0026] [0027] [0030], [0295] through to [0302], [0308] through to [0734]), using the paragraph numbers allocated by the USPTO).

Conversely, I am not attempting to claim ownership of any of the existing well-known concepts. This includes the terms Transaction Time (also called Y Time), Valid Time (also called X Time), Chronons and the 'moving point now', which are all prior art.

I will proceed with stating one of many definitions of a Temporal Database. A Temporal Database is simply a database that holds Temporal Data, and Temporal Data is simply data that is in some way defined with reference to, or constrained by, the dimension of Time.

In the real world, we are all creatures constrained by time. We live in the moment, now, and can only imagine the future or indeed remember the past.

However, in the business world (and in fact with any enterprise storing data), the data we manipulate within our systems exists within a logical world, not constrained by the physical dimension of time. That is to say we often store information that will become true in the future and record information that was true in the past. We write programs that use that data to provide, for example, audit reports and forecasts. In fact the business world has very sophisticated requirements with respect to the manipulation of data within the context of time and always has done.

There are numerous examples of this:

Consider, say, a stock trading system that not only needs to record when a trade was made, but also needs to provide a facility to rerun or model trading patterns that occurred in the past. This could help to discover unusual patterns of trading, which in hindsight could point to insider trading.

Retailing systems may need to set a future price or future discount for a product and until that time occurs offer a different price or discount. Customer Relationship Management (CRM) systems need to record future changes of address and addressee names so that correspondence will automatically switch to the correct contact and contact point when required. Compliance and disclosure legislation requires that a business can prove what information was known in the past - to show that decisions were legally made at the time; despite subsequently recorded facts showing that the decision was in error. Sophisticated requirements have always existed, the question is can the existing (relational) technology address the requirements?

To answer this I need to describe the database developments which took place in the latter part of the 1970s decade. This was a time when hierarchical databases were the generally adopted database technology. An accomplished mathematician called Dr E F Codd (who worked for IBM) was the first person to realise that the concepts of mathematical Set Theory could be applied to database data management for technical and therefore commercial benefit. As a result Dr E F Codd defined his famous 12 relational principles upon which all relational databases are based. This revolution caused the rapid demise of the hierarchical model in favour of the relational model which we have all been using since.

Paragraphs discussing this in my patent include paragraphs [0013] [0014] [0015] [0019].

But, although relational technology is powerful, there is a problem. The problem is that the 12 relational principles do not inherently provide support for the crucial dimension of time. This is indeed ironic as time is probably the most important requirement! Relational technology in its purest form is a CURRENT VIEW model. The only temporal support it offers is the ability to commit data at a particular transaction time and the ability to store a date datatype. If new facts emerge the existing data must be overwritten. This previous data is then lost unless the database designer and/or application coder decides to perform additional processes to store that data.

In relational technology it is common practice for designers to provide these additional coping mechanisms because the business cannot tolerate a system that only stores CURRENT knowledge. There are many examples of such coping mechanisms e.g. History tables, Audit Tables, additional columns to hold FROM dates and TO dates etc.

Some relevant paragraphs in my patent application are [0007] [0008].

As designers successfully cope with this lack of temporal support, so the business pushes the technology even harder.So, for example, a Datawarehouse built on relational technology nearly always defines its facts with respect to the 'slowly changing dimension of time' as well as other dimensions such as geography or organisation.

More recently, business has pushed the relational technology harder still by asking for systems that provide support for legal disclosure and Sarbanes-Oxley (SOX) compliance.

Designers have responded (or coped) by providing solutions such as Information Lifecycle Management (ILM).

In fact application programmers and designers are so resourceful and so used to coping that they just accept the massive effort required to compensate for this lack of temporal support as if there were no alternative option. And until my invention there was no alternative.

The alternative inventive option though is a Temporal Database and its existence can fundamentally revolutionise the way in which systems are constructed. Temporal technology provides the temporal support that is currently missing from relational technology. This would mean the end of the massive coping effort by the IT system kernel and bespoke programs written to cope with relational rules over time. This frees up the time of the designers so that they can concentrate on business or enterprise requirements rather than on needing to accommodate how the technology will cope with the requirements.

Additionally, because temporal support is inherently provided by my new database it means that it is much more reliable, efficient, accurate and sophisticated because it does not rely on the individual skill of the software implementers and is not reinvented for each system developed.

One final thought before I explain the model specified in my patent application. If we are so good at coping with time changes, and have been using relational technology for 25 years or more, what is the motivation for changing our approach? Why not just continue with current technology? Well, the best way I can think of convincing someone that another approach is worth considering is to refer to a well-established mathematical technique. This technique states that it is possible to solve intractable or extremely difficult problems in one area of mathematics (lets say Algebra) by posing the question in another form (lets say Trigonometry). When the difficult problem is posed in this new form then it can become trivial to solve. Once solved the answer can be translated back to the original form.

If relational technology is considered as one 'form' and Temporal technology is considered as another 'form', then the problems that have been "coped" with for the last 25 years can be resolved easily using Temporal technology or more specifically, the invention contained within the patent.

Temporal technology could revolutionise the way relational systems are built and provide levels of sophistication that could not be achieved with non-temporal relational technology. It is a revolution from the bottom up and has the potential to affect nearly all aspects of systems including configuration management, version control, archiving, backup strategies and stateless internet connections.

See patent paragraphs [0735] through to [0818].

As mentioned previously, the benefits of temporal databases are discussed in prior art, but Temporal Databases, although highly desirable do not exist because as of yet there has not been a compelling, robust and implementable design or model for temporal technology.

Alternatively, you could say that we do not have the ability to restate our relational problems in a modified way and so we are left merely to cope with (or in some cases fail to solve) difficult business requirements with relational technology.

That is why it is necessary to build Temporal Databases. This patent specifies an inventive and new design, called Temporal Management Normal Form (TMNF) or the Summary Position, along with a set of well defined processes and rules which enables an implementation to be built.

See patent paragraphs [0031] through to [0294].

Prior art defines four levels of classification of temporal databases. The criteria upon which the classification depends is simply the inherent support provided for temporal data. That is to say, what features exist 'out of the box' before designers add coping mechanisms eg history tables, to make up for the lack temporal support

Level 1 is concerned with transaction time and is sometimes called a Snapshot Temporal Database. Level 1 inherently supports the ability to store or commit a current view of data at a particular transaction time. Most commercially available relational database products can be classified as being a Snapshot database. Lomets patent is concerned with transaction time and so helps to enable level 1 databases.

Level 2 is concerned with transaction time and is sometimes called a Rollback Temporal Database. Level 2 provides level 1 support plus an inherent ability to view previous states of data with respect to (or as at) a particular transaction time. Some commercially available relational databases already provide level 2 support eg Oracle Flashback Query.

Level 3 is concerned with valid time and is sometimes called an Historical Temporal Database. Level 3 inherently supports the ability to view previous states of data with respect to (or as at) a particular valid time. The ability to offer this kind of support is unavailable in current commercially available databases. It is left entirely upto the ingenuity of application designers to provide this support if they can.

Level 4 is concerned with both transaction time and valid time and is sometimes called a Bi-Temporal Database. Level 4 inherently supports the ability to view previous states of data with respect to (or as at) a particular valid time and transaction time.

Temporal Normal Form (TMNF) or Summary Position together with the processes defined in this patent application is the inventive feature that provides the inherent support required for a level 4 temporal database.

<u>The Examiner's Objections</u>

I do not understand some of the Examiner's objections. I appreciate the Examiner is saying that his objections are perfectly clear, and that I should read the MPEP. I have read the MPEP sections that relate to the Examiner's objections and believe that this response satisfies them. They did not clarify the Examiner's objections.

<u>35 USC 112 claim rejection of claims 12, 15, 16, 17, 18, 20 and 21.</u>

This objection is surprising to me. The amendments to these claims basically add the features of original claims 3 and 4 into the previous claims 12, 15, 16, 17 and 18.

No 35 USC 112 objection was made against original claims 12, 15, 156, 17 and 18. Objection under that section was made against independent claim 1 and independent claim

14. The fact that there was no objection under this enablement ground against independent claims 12, 15, 16, 17 and 18, but there was against independent claims 1 and 14 was the reason why claims 1 and 24 were deleted, without prejudice, and claims 12, 15, 16, 17 and 18 were retained.

I do not understand how adding narrower features that were in claim 3 and 4 can make a claim that was previously enabled across its wide breadth non-enabled for a narrower breadth.

Has the Examiner simply changed his mind?

The Examiner encouraged me to read the MPEP, and I thank him for that suggestion. The MPEP says that the burden of proof on non-enablement lies with the examiner and a prima facie case needs to be laid out by the examiner – he needs to explain why the specification is not enabling . The examiner has not done that. He has simply said that it does not meet 35USC 112 and recited the last part of the claim. He has not said why he thinks that the skilled man cannot do the "performing updating and deleting operations " as claimed. He just says that he cannot. Why not ?

Since this is the first time the examiner has objected to these features – previously they were in dependent claims and no objection was raised - it seems unfair to make the rejection a final rejection.

In any event, the Examiner has indicated that claim 21 recites "performing updating and deleting operations on the values of attributes by inserting into the database a new inset summary position with a new particular transaction time without altering any existing summary position, logical update and delete of valid start time and valid end time for a value of an attribute achieved by inserting the new summary position with a different valid start time or valid end time that is to be true for the particular transaction time".

The Examiner indicated over the telephone that I should explain how the skilled man can perform each and every feature of the claims.

I will start with claim 21:

21. **A method of updating and deleting values of attributes in a database,** *[this is technically clear to the skilled man, and achievable using the method steps that follow]* **the database having stored in it a summary position in which attributes have associated with them a valid start time from which the value of the attribute will hold true and a valid end time up to which the value of the attribute will hold true,** *[a computer memory (readily understood and available) will have data entries previously input (e.g. by typing them into a table/database) of valid start and end times associated with an entry for the value of an attribute (this entry also entered into a database). A skilled person can produce this database]* **and a transaction time associated with the summary position at which transaction time the valid start time and valid end time of the value of the attribute are true,** *[again the transaction time is an entry in the database associated with the summary position entry and linked/associated with the value start and value end times. The transaction time is typically the time (system clock time) at which the valid start and valid end times are true. The skilled person readily understands that, after reading the present patent application, the valid start and valid ends times of the value can be different at different transaction times and so knows that the valid time start and end times for the particular value of the attribute are applicable for one or more particular points in transaction time. Again, data entry into a database, and the skilled person will find this technically achievable once the desire to do it has been explained – linking a valid start time, valid end time, and a system clock – created transaction time at which the valid start time and valid end time were entered into the database is achievable]* **the method comprising performing updating and deleting logical operations on the values of attributes by inserting into the database a new insert summary position with a new [particular] transaction time without altering any existing summary position,** *[the skilled person knows how to create a summary position - discussed above. The new inset summary position is a new summary position inserted/loaded into the database (relating a value with a valid start time and a valid end time, for a new/different point in system-generated transaction time. That is to say, the new entry into the database specifies a different start and/or valid time for the value already in the database, but not by changing the existing entries – instead by writing a new entry, at a later point in system clock time (transaction time)* **logical update and delete of valid start time and valid end time for a value of an attribute being achieved by inserting the new summary position with a different valid start time or valid end time that is to be true for the new [particular] transaction time.** *[This defines how the update of a value is*

achieved in the method – by, effectively, saying that when I query the database and run the query effective at transaction (system) time, the value of the attribute I will retrieve from the query depends upon the value entered in the database that I have linked with the transaction time effective for the query (and that is valid for valid time at the query point). For example, to change the value of "Number of Employees" from three to four, on 01 August 2006 because a new person started to be employed, the existing summary position may say:

Attribute: Number of Employees
Value: 03
Valid Start Time: 01 January 2002
Valid End Time: 01 January 2050
Transaction Time: 20 May 2006 (let us say that the entry was made in the database on 20 May 2006)

The existing entry is left in memory unchanged.

The new summary position inserted into the database as a new entry says:

Attribute: Number of Employees
Value: 04
Valid Start Time: 01 August 2006
Valid End Time: 01 January 2050
Transaction Time: 01 August 2006 (if the entry was made on 01 August or, say,
10 August 2006 (if the entry was made on 10 August 2006)

When a query is made of the database, the query specifies what transaction time/system time is to be used for the query: i.e. when do I want to pretend that the query was made? What answer would I get if I had asked the query on 05 August 2006? It is now December 2006, but I want to know what answer I would have got on 05 August. Well, it all depends upon when the new summary position was altered in the database. Let us take the example that the database had the new summary position added on 01 August (before the effective transaction query date of 05 August). Then, the query would find the new summary position effective as of 01 August and return the answer "4".

But, if the database had not really been added to until 10 August, then the effective query pretending the system time at the time of query was 05 August would find the transaction time to be used was not that of 10 August – too late – but instead the preceding summary position associated with the last-entered transaction time that is before the effective *transaction query time – the summary position as of 20 May 2006, and would return the answer "3".*

Two different answers to the same question, depending upon the transaction time used to query the database, and depending upon when in the transaction/system time the database was really added to.

I can look at back at the true position the system knew about in the past because the old data is still there – including start and end times for values and attributes.

If I had ***changed*** *the start or end time of the previously existing value, I would not have been able to query a historical position. For example, if on 10 August 2006 when I update the database I* ***change*** *the existing end time for the value "3" from ending in 2050 to ending on 31 July 2006, and enter a new value for "Number of Employees" as being "4" from a valid start time of 01 August 2006 to and end term of 2050, I cannot go back and get the answer "3" when I pretend to query the database on 05 August – even though the query would really have returned "3" as the answer, the computer now tells me that it would have said "4".*

Because I do not alter the existing start and end valid times for values of attributes, the query "What answer would I have been given if I had asked at transaction time XYZ?" is faithful to reality, rather than not being actually correct.]

It is not understood why the Examiner thinks that this is non-enabled. He felt that claims 3 and 4 were enabled in the last Official Action. If it makes a difference, I am happy to go back to the exact language of claims 3 and 4.

Having said the above, I see from the MPEP that there is a preference to tie in "real world" structure so as to be more concrete. The amendments now made to Claim 21 (and not mentioned in the above analysis) are to specify that the method is a <u>computer-implemented</u> method. That the database is held in <u>computer-readable memory</u>. That the method is performed by a <u>computer processor</u> to insert into <u>computer-readable memory</u> the new summary position leaving the previous summary position still in the <u>computer memory</u>. All of these are specific and definite technological terms. I have deleted [particular] since the Examiner did not appear to like it and the claim reads well without it. There is, of course, disclosure explicit, important, and inherent, for such amendment throughout the specification. The amendment does not raise any new issues – since it is simply a clarification of the existing claim to more clearly, demonstrate that 35 USC 112 and 35 USC 101 are satisfied.

The claimed invention produces, achievably by a skilled man, a substantial, specific, and credible result – a database and database operating method that produce more truthful answers to queries which relate to what things were/are at different points in time for values that change over time. It does this in a new and non-obvious way by insert-only operations, including on timestamps for start and end times of values.

I note that the MPEP says "when functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases...."

It also says "a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships.... and is this statutory". The amendments to claim 12 are in response to these concepts. Again, the amendments to claim 12 are not introducing new issues: they merely clarify the previously existing database structure. The amendments have basis explicit, implicit, and inherent in the text and drawings of the original application.

The Examiner later goes on to object that claims 2, 7, 10 to 12 and 14 to 20 are anticipated by Lomet (US 6 754 657). The skilled person knows Lomet, and can enable Lomet/can do Lomet. It is logically inconsistent to say that the claims are not enabled by the specification, but are actually **disclosed** by Lomet. I do not think that the examiner is saying that Lomet is non-enabled and should not have been granted. If the skilled person can do Lomet, why can he not do my invention, after having read the patent application? (Clearly, he cannot think of doing my invention before he reads my patent application because the invention is non-obvious).

There is a lot of support for claim 21 being enabled in the specification, namely:-

the discussion of the various rules that might be used in one embodiment, paragraphs [0046] to [0085];
a discussion of insert-only functionality being used to create the same effect as other logical commands , paragraphs [0120] to [0128];
the diagrammatic representation of the invention I the many drawings of Figures 1 to 87;
the discussion of the invention model in paragraphs [0310] to [0344];
detailed embodiment discussion exists in paragraphs [0394] onwards, including built in instructions for Action-Result Diagrams [0570]. Detailed guidance on how to construct the database. The skilled man can perform the invention.

Furthermore, 35 USC 112 is a factual issue. I enclose a draft Affidavit from a Mr. John Callaghan swearing on Oath that he has read the patent application and that he has no doubt that he could made a suitable database and perform the methods disclosed, and that he actually did so. The signed original Affidavit will follow shortly.

I submit that the Examiner should not put his own opinion ahead of evidence filed. The Examiner himself did not think it was an objection in the first Official Action because he was not concerned that it was non-enabled.

I propose to discuss the other independent claims, and the 35 USC 112 objections against them, later.

REJECTION UNDER 35 USC 101 – NO UTILITY

The Examiner argues that the invention as a whole is not useful and does not accomplish a practical result.

A process is patentable under the MPEP , and a claim to a method which recites one or more acts to be performed defines a process. The claimed process results in a physical transformation for which a practical application is disclosed – the memory in which the database is stored is changed by each insert , and the result of performing a query operation is the creation of a return signal from the database having data in it for display or storage in computer memory– a real thing.

The practical result of claim 20 is discussed above – getting a different answer when the bi-temporal (valid time and transaction time) database is queried than the prior art. Getting a result/answer to a query now, today, which asks "what would the answer be to this query if I had asked it at a specified time in the past" is a useful, concrete, real-world application.

The practical result of the invention of claim 20 is better and more truthful answers to some kinds of queries.

Also, another practical, useful, result of updating and deleting values of attributes in a database in accordance with the present invention is that the database and database querying system is more easily modified in the future to enable it to be queried for new things not originally planned-in to the functionality of the database. In the prior art, it is necessary to have lots of detailed relational rules reviewed, and possibly altered, to enable a new value to be queried, or to take into account functional changes. In the present invention, this is unnecessary and the future modifications amount, typically, and largely to, populating a database with data for the new attribute, rather than code-level programming. This future flexibility is very useful.

Examples of real world usefulness of the invention are given throughout the patent application, and in particular at paragraphs [0735] to [0818], and [0037], [0030], [0041], [0042], [0077], [0147], [0150], [0151], [0310] and [0453].

I will discuss claims 12, 15, 16, 17, 18 and 20 in relation to 35 USC 101 later.

<u>Claim Rejection 35 USC 102</u>

The Examiner objects to claim 21 as being anticipated by US 0754657 Lomet. The Examiner argues that Lomet discloses insertion of a new summary position with new start valid time and end valid time entries without altering any existing summary position. I think that Lomet does not show this and that the document has been misconstrued.

The Examiner refers to column 1, lines 10-25. These discuss the general desire to have "transaction time support", but not how it is to be achieved. No reference to not altering or deleting data relating to pre-existing positions.

The Examiner also refers to column 1, lines 60-67. These lines read "A common query in a database having transaction time support is termed a time slice. A time slice asks for the set of data items that were current at some past time t. The time slice query is answered by finding each data item d having a start time d. TT on or before the past time t requested and an end time d. TT that was on or after the time t requested. The time slice at time t returns the state of the database that was current at the past time t."

This paragraph discusses the background art, not the invention of Lomet, - it simply says that there is a start and end time – nothing about that they cannot be changed/are not altered.

It is not understood how the Examiner finds a disclosure of the claimed concept of update and delete achieved by insert only entries, with no alteration of existing summary positions in the cited paragraphs (including the start time and end time (in valid time) of values of attributes).

Lomet itself, as has been previously said by the Examiner in the Official Action of 13 April 2006, "does not disclose wherein changes to said values are achieved by inserting new entries for linked transaction time, valid time start time, valid time end time, and associated attribute value over a hereby specified valid time period, there being no actual delete operation", page 8 final paragraph, of the Examiner's previous Official Action. I do not understand how the Examiner has changed his mind on this point of what Lomet discloses. I agree with his first position and disagree with his later position. Lomet does alter previously recorded summary positions – see column 7, lines 12 to 22.

"The disclosed system and method varies the manner in which timestamps are assigned based on a request for current time by a transaction. The disclosed system and method sets an initial transaction timestamp lower limit. During a first phase of transaction execution, the lower limit is updated based on reading and writing of data. If a data item is read, the timestamp lower limit is updated to a time that is at least as large as the time the data was written. If a data item is being written the timestamp lower limit is updated to a time at least as large as the time the data was previously read or written." Changing existing timestamps/start or end valid times. This is in direct opposition to leaving them unchanged and simply writing a new entry.

Similarly, column 7, line 66 to column 8, line 3 of Lomet reads: "the read timestamps stored in the hash table are updated to ....". Changing existing entries in the database. Not what is claimed at all.

Claim 21 is novel and inventive over Lomet.

Turning now to **claim 12**.

35 USC 112 objection – not enabled

This claim is to a database. I have already gone through how the skilled man can perform a method using the database. X-time in the claim is valid time, and Y-time is transaction time (or system clock time). I do not understand what it is that the Examiner says cannot be done? The application carefully talks a reader through how to create the claimed database. In our telephone conversation I had the impression that the Examiner did not like the term "particular" transaction time or "appropriate" time. No such terms appear in claim 12. They appear in the original abstract, and the original claim 1, but not in the last-presented, or present, claims.

35 USC 101 – inutility

Claim 12 claims a database. The Examiner confirmed over the telephone that many thousands of databases are patented by the USPTO, and that databases do have utility. I cannot see how the Examiner feels that my database has no utility, when all the granted database claims do have utility. The database is useful in the real world. Many examples of situations where by database is better than prior art databases are given in the application (e.g. those in paragraphs [0735] to [0818] to name a few).

35 USC 102 – lack of novelty

Lomet quite clearly does not have a database which permits insert only operations. Lomet alters existing entries, as true delete and updates, not retaining all previous entries.

Claim 15

This claim claims a method of holding data in a database.

35 USC 112 – non-enabled

I just do not see how this invention is not enabled. In the previous Official Action the Examiner accepted that it was enabled – there was no objection to claim 15 under 35 USC 112. The Examiner has changed his mind and I do not understand why.

The Affidavit of John Callaghan agrees that the claimed invention is enabled.

35 USC 101 – no utility

Again, claim 15 was not previously objected to under this ground in the Examiner's earlier Official Action. I submit that the utility of claim 15 is clear. The benefits of holding the data in a database in the claimed way is clear from the patent application.

35 USC 102 – lack of novelty

Lomet simply does not disclose a method in which <u>all</u> logical update and delete operations are performed by inserts – it specifically says to modify existing timestamps. Clearly not an insert operation.

Claims 16, 17, 18 and 20

These are independent claims to

- A method of modifying the value of an entity in a database;
- A method of modifying the value of an entry in a database;

- A method of modeling changes in values of attributes in time in a database; and
- In a temporal database system ...... an improved method for granting access during the modification of the information in a database.

35 USC 112 – non-enabled

As will be apparent, I simply do not see any basis for the Examiner objecting on this ground. All of these claims are enabled by the disclosure. John Callaghan, an independent programmer, agrees that they are enabled.

It is not at all apparent which feature (or features) is not enabled, according to the Examiner.

35 USC 101 – no utility

Again, for all of the claims, the utility is clear and specifically discussed in the application.

The use is to enable a better database to be made, and to enable future changes to allowable queries to be made more easily.

35 USC 102 – lack of novelty

The Examiner's discussion of claims 16, 17 and 18 fails to discuss the claimed insert only operation, with logical update and delete being achieved by an insert operation, with the previous database entries remaining unchanged (as required by the amendments to claims 16 and 17, and as always required by claim 18).

Lomet discloses altering existing timestamps: not what is claimed at all.

In relation to claim 20, the Examiner alleges that Lomet discloses insert-only operations performing alter and/or delete, and refers to Figure 8. I cannot see anything of the sort in Lomet. It clearly discusses changing existing entries. Figure 8 specifically shows update and delete as being separate to and different from insert.

New Claim 22

This is similar to old claim 21, but with more structure recited, more physical embodying. Reading the MPEP suggests that this may be helpful, although I firmly think that the existing database claim is to a physical thing – apparatus , and that the method claims recite steps to take and are processes which alter the state of a physical memory and/or the display on a screen/signals representative of a response to a query. The MPEP suggests that they should not be objected against.

Response to Arguments

The Examiner has misunderstood my earlier arguments. I do refer to one difference between Lomet and my invention, in the third paragraph of page 1 of the Remarks, in my last response as relating to when the timestamps are set. However, that is only ten lines in my overall arguments. None of the remaining five pages of arguments rely upon that. In particular, the third paragraph of page 2 of my Remarks said, "It is not obvious to keep the timestamp for start and end times of a value unchanged ……". The top of page 3 of the Remarks says "no motivation to create additional summary positions for each change." The middle paragraph on page 4 of the Remarks discussing claim 21 discusses "Logical delete or update of start/end timestamps for values is achieved by an insert operation only process, the previous timestamps remaining unaltered."

All of these arguments relate to features that are in the claims, and that are not in the prior art.

The last two pages of the previous Remarks also focus on not updating existing timestamp values.

It is not seen how the Examiner can possibly say that the features upon which I rely are "i.e. the timestamp for the start time is set at the time of writing the value of the attribute……". I do not rely on that. I rely on what is in the claims (and the same features are not in Lomet, as discussed).

CONCLUSION

I believe that the claims are patentable. I note that the MPEP says that whenever practicable , the USPTO personnel should indicate how rejections may be overcome and how problems may be resolved. If the examiner feels that the introduction of more “memory”, “processor”, “database held on computer-readable media”, language would help them I would be very pleased to include that.

If the examiner feels that claiming a computer or computer network programmed to perform the method would help, then I would be pleased to change the claim language.

Respectfully submitted

Luke Porter

Luke L.M. Porter
46 Margaret Grove
Harborne
Birmingham B17 9JL
Great Britain.

Dated: 1st December 2006